What is claimed is:

1. A method comprising:

separately performing a first one or more arithmetic operations and a second one or more arithmetic operations, wherein the second arithmetic operations indicate if the first arithmetic operations cause a carry condition; and

providing one or more results of the first and second arithmetic operations.

2. The method of claim 1 further comprising:

executing a first instruction to perform the first arithmetic operations; and

executing a second instruction to perform the second arithmetic operations.

3. The method of claim 2 wherein the first and second instructions are executed in parallel.

4. The method of claim 2 wherein the first instruction and the second instruction correspond to a very long instruction word packet (VLIW).

5. The method of claim 2 wherein the first instruction corresponds to an add or subtract instruction and the second instruction corresponds to a generate carry instruction or a generate borrow instruction.

6. The method of claim 2 further comprising:

executing a third instruction to perform the first arithmetic operations and a fourth instruction to perform the second arithmetic operations, wherein the first and second instructions operate on a first and second operands, which are the lower portions of a first and second values, and the third and fourth instructions operate on a third and fourth operands, which are the upper portions of the first and second values.

7. The method of claim 1 wherein the second arithmetic operations determine if the first arithmetic operations will require a carry, indicate a carry bit, and update a result of the first arithmetic operations accordingly.

8. The method of claim 2 wherein the first instruction indicates an opcode, a first and second operands, and a destination for a first result, and the second instruction indicates an opcode, a first and second operands, and a destination for a second result.

9. The method of claim 1 wherein the second arithmetic operations determine if the first arithmetic operations will require a borrow, indicate a borrow bit, and update a result of the first arithmetic operations accordingly.

10. The method of claim 1 embodied in a computer program product that is encoded on one or more machine-readable media.

11. A method comprising:

decoding an instruction packet;

performing an arithmetic operation in accordance with the decoded instruction packet with a first and second operand;

determining, separately from performing the arithmetic operation,

if the arithmetic operation results in a carry if the arithmetic operation is an add operation, or

if the arithmetic operation results in a borrow if the arithmetic operation is a subtract operation; and

indicating the carry if the carry result is determined and the borrow if the borrow result is determined.

12. The method of claim 11 wherein the instruction packet is a very long instruction word packet.

13. The method of claim 11further comprising executing the determining and the performing in parallel.

14. The method of claim 11 further comprising:

performing the arithmetic operation with a third and fourth operand, wherein the first and second operands are respectively lower portions of a first

and second value and the third and fourth operands are respectively upper portions of the first and second values; and

updating the results of the arithmetic operation with the determined carry result or borrow result.

15. The method of claim 25 further comprising concatenating the result of the arithmetic operation of the first and second operands with the results of the arithmetic operation of the third and fourth operands.

16. A processor that supports separately executable instructions that perform one or more add type operations and one or more carry operations that correspond to the add type operations.

17. The processor of claim 16 that also supports separately executable instructions that perform one or more subtract type operations and one or more borrow operations that correspond to the subtract type operations.

18. The processor of claim 17 that supports parallel execution of the separately executable instructions.

19. The processor of claim 16 that decodes the separately executable instructions from a very long instruction word packet.

20. An apparatus comprising:

an arithmetic logic unit; and

means for separately performing add type operations and carry operations.

21. The apparatus of claim 20 further comprising means for separately performing subtraction type operations and borrow operations.

22. The apparatus of claim 21 further comprising means for performing the subtract operations in parallel with the borrow operations.

23. The apparatus of claim 20 further comprising memory and one or more processors.

24. The apparatus of claim 20 further comprising means for performing the add operations in parallel with the carry operations.

25. A computer program product encoded on one or more machine-readable media, wherein the computer program product comprises:

a first subsequence that includes a first instruction to perform add type operations on one or more operands and a second instruction to indicate if execution of the first instruction will result in a carry; and

a second subsequence that includes a third instruction to perform subtraction type operation on one or more operands and a fourth instruction to indicate if execution of the third instruction will result in a borrow.

26. The computer program product of claim 25 wherein the second instruction also sets a bit in a designated location to indicate a carry result and the fourth instruction also sets a bit in a designated location to indicate a borrow result.

27. The computer program product of claim 25 wherein the instructions include an opcode field, one or more source operand fields, and a destination field.

28. The computer program product of claim 25 wherein the first and second instructions are executable on separate processors, and the third and fourth instructions are executable on separate processors

29. The computer program product of claim 25 wherein the first and second instructions are executable by separate functional units in parallel, and the third and fourth instructions are executable by separate functional units in parallel.

30. The computer program product of claim 25 wherein the first and second subsequences each include a very long instruction work packet.